Computer Assisted Methods in Engineering and Science, **21**: 223–232, 2014.


# Efficient Markov chain Monte Carlo sampling for electrical impedance tomography

Erfang Ma

*Department of Mathematical Sciences*
*Xi'an Jiaotong Liverpool University*
*Renai road 111, Suzhou, China*
*e-mail: erfang.ma@xjtlu.edu.cn*

This paper studies electrical impedance tomography (EIT) using Bayesian inference [1]. The resulting posterior distribution is sampled by Markov chain Monte Carlo (MCMC) [2]. This paper studies a toy model of EIT as the one presented in [3], and focuses on efficient MCMC sampling for this model.

First, this paper analyses the computation of forward map of EIT which is the bottleneck of each MCMC update. The forward map is computed by the finite element method [4]. Here its exact computation was conducted up to *five* times more efficient, by updating the Cholesky factor of the stiffness matrix [5]. Since the forward map computation takes up nearly all the CPU time in each MCMC update, the overall efficiency of MCMC algorithms can be improved almost to the same amount. The forward map can also be computed approximately by local linearisation, and this approximate computation is much more efficient than the exact one. Without loss of efficiency, this approximate computation is more accurate here, after a log transformation is introduced into the local linearisation process. Later on, this improvement of accuracy will play an important role when the approximate computation of forward map will be employed for devising efficient MCMC algorithms.

Second, the paper presents two novel MCMC algorithms for sampling the posterior distribution in the toy model of EIT. The two algorithms are made within the 'multiple prior update' [6] and the 'delayed-acceptance Metropolis-Hastings' [7] schemes respectively. Both of them have MCMC proposals that are made of localized updates, so that the forward map computation in each MCMC update can be made efficient by updating the Cholesky factor of the stiffness matrix. Both algorithms' performances are compared to that of the standard single-site Metropolis [8], which is considered hard to surpass [3]. The algorithm of 'multiple prior update' is found to be *six* times more efficient, while the delayed-acceptance Metropolis-Hastings with single-site update is *at least twice* more efficient.

**Keywords:** electrical impedance tomography, Bayesian inference, Markov chain Monte Carlo.

## 1. Introduction

Electrical impedance tomography (ETI) concerns inferring the distribution of electrical property over a body, with measurement of voltage and current on the boundary of the body [9]. EIT has many applications in various fields, and has been extensively studied during the past years [10].

This paper discusses the reconstruction problem of EIT using Bayesian inference with MCMC. The Bayesian approach to EIT was pioneered in [1, 6]. The Bayesian approach results in a posterior probability distribution of an unknown electrical distribution. With samples of this distribution, we not only can estimate the unknown electrical distribution, but also quantify the error of this estimate. Drawing samples from this posterior distribution is however a non-trivial task and has to rely on the Markov chain Monte Carlo (MCMC) method [2]. Though many MCMC algorithms work for sampling this posterior distribution, few are efficient enough for practical use [3]. In [3], performance of various MCMC algorithms is compared on sampling a posterior distribution for

a toy EIT model. It was concluded that the classical single-site Metropolis [8] is actually hard to surpass for this application, and more efficient MCMC algorithms are required in order for the Bayesian approach to EIT to be used in practice.

This study follows the work in [3], and focuses on efficient MCMC sampling for EIT. We speed up the computation in each MCMC update by updating the Cholesky factor of stiffness matrix in successive calls to the forward map of EIT. We also employ two variants of Metrpolis-Hastings schemes [11], to make novel and efficient MCMC algorithms. The paper is organized as follows. In Sec. 2, we introduce the toy EIT model that this paper focuses on, the posterior probability distribution resulting from the Bayesian approach to this EIT model. MCMC algorithms will be employed for sampling this distribution. In Sec. 3, we speed up the computation of successive calls to the forward map of EIT in the context of MCMC. This is done by updating the Cholesky factor of the stiffness matrix in a finite element formulation. We also make local linearisation of forward map more accurate, by introducing a log transformation into this process. In Sec. 4, two novel and efficient MCMC algorithms are proposed and employed for sampling the posterior distribution in our toy EIT model. Finally, we conclude the findings and discuss the future work.

## 2. A toy model of EIT

The EIT model discussed in this paper is shown in Fig. 1. In this figure, a phantom conductivity $x$ is distributed over a square region $\Omega$. Assuming that the region is divided into a $24 \times 24$ grid of squares, the true conductivity is constant on each square, and is either three (white colour in the figure) or four (black colour in the figure) in some arbitrary unit. To infer this conductivity distribution, sixteen point electrodes, numbered 1–16, have been evenly placed on the boundary $\partial\Omega$ of the region. Current is injected through one of these electrodes and extracted uniformly through the boundary $\partial\Omega$. The resulting voltage at all of the electrodes is measured. This procedure is then repeated for sixteen times, with each electrode being the injector of current for once. In the end, sixteen sets of measured voltages at all the electrodes are obtained, and they are denoted as $u_E$.

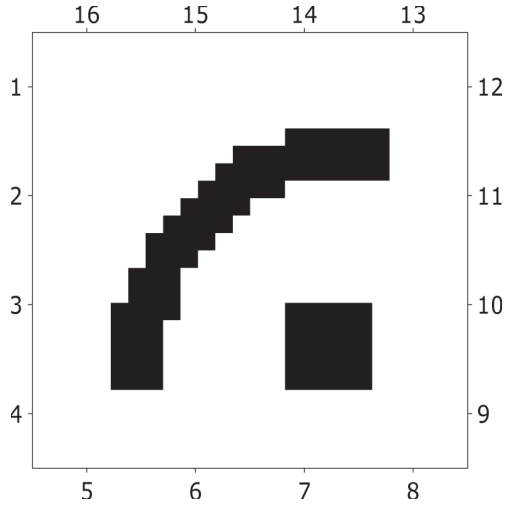


**Fig. 1.** The toy EIT model. Phantom conductivity is distributed over a square region. White and black represent conductivity to be three and four in some arbitrary units. Sixteen point electrodes denoted as 1,2,. . . 16, are evenly placed on the boundary of the region. Through these electrodes currents are injected and the resulting voltages are measured.

The measured voltages $u_E$ at all the electrodes depend on the conductivity distribution $x$ over the region. This dependence $u_E(x)$ is the forward map of EIT, given that the injected current

is of a fixed pattern. Evaluating $u_E(x)$ for some $x$ involves solving the following boundary value problem (BVP):

$$-\nabla\cdot(x(s)\nabla u(s))=0,\quad s\in\Omega, \tag{1}_1$$

$$-x(s)\frac{\partial u}{\partial n}(s)=j(s),\quad s\in\partial\Omega, \tag{1}_2$$

$$\int\limits_{\partial\Omega} j(s)\mathrm{d}s=0, \tag{1}_3$$

$$\int\limits_{\partial\Omega} u(s)\mathrm{d}s=0, \tag{1}_4$$

where $u(s)$ denotes the potential at spatial point $s$, $(1)_4$ represents a way of selecting the potential reference that ensures the solution of this BVP is unique, $j(s)$ is the current pattern through the boundary $\partial\Omega$ of the region, $j(s)$ has to satisfy, Eq. $(1)_3$ because of the current conservation. In this study, $j(s)=\delta(s-e_k)-1/|\partial\Omega|$. $\delta(s-e_k)$ is a Dirac delta function, $e_k$ denotes the location of the $k$-th point electrode on the boundary $\partial\Omega$, $|\partial\Omega|$ is the length of the boundary. For a conductivity distribution $x$, we solve this BVP for $u(s)$ which contains $u_E$. This BVP is solved by the finite element method (FEM). In FEM, the region has been divided into $24\times 24$ squares. So the conductivity $x$ is of $24\times 24$ components. It is undetermined to infer $x$ with only $16\times 15$ measurements. However, the Bayesian approach can overcome this issue by introducing a prior distribution for $x$.

In the Bayesian approach to our EIT model, the conductivity $x$ is considered to be random. As in [3], an example of the posterior distribution resulting from the Bayesian approach to this model could be:

$$\pi(x|u_E^{\mathrm{noise}})\propto\exp\left\{\beta\sum_{i\sim j}v(x_i-x_j)\right\}I_{[2.5,4.5]^n}(x)\cdot\exp\left\{-\frac{1}{2\varsigma^2}\|u_E^{\mathrm{noise}}-u_E(x)\|^2\right\}. \tag{2}$$

Here $i\sim j$ is defined to be a horizontal or vertical nearest neighbour as stated in [3], $v(\cdot)$ is the tricube function of [12]:

$$v(t)=\begin{cases}\dfrac{1}{s_0}\left(1-|t/s_0|^3\right)^3, & \text{if } |t|<s_0,\\ 0, & \text{otherwise,}\end{cases} \tag{3}$$

and $I_{[2.5,4.5]^n}(x)$ is an indicator function restraining every component of $x$ to be within the range $[2.5,4.5]$, $n=24^2$ because $x$ is a $n$-dimensional vector. $u_E^{\mathrm{noise}}$ is the voltage measured at the electrodes that is contaminated with additive noise, $\varsigma$ is the level of noise and $\beta$ is a regularization parameter. Here it is set to be 0.5 as in [3]. Samples from this posterior distribution can be used to produce estimate for the unknown conductivity distribution. To draw samples from this distribution, the MCMC method is employed.

## 3. Forward map computation

### 3.1. Exact computation

As indicated in Sec. 2, evaluating the forward map $u_E(x)$ consists of solving the BVP (1) using the FEM [4]. The FEM results in a sparse and positive system of linear equations:

$$Ku=f. \tag{4}$$

Its solution is the potential $u$ over the region that certainly contains $u_E(x)$. In (4), the stiffness matrix $K$ depends on the conductivity $x$ as follows:

$$K = \sum_{i=1}^{n} x_i K_i + \lambda c c^T. \tag{5}$$

Here $K_i$ for $i = 1, 2 \ldots, n$ are all sparse and symmetric matrices. They are all built on the same local stiffness matrix, and depend only on the geometry of the mesh and the element function employed, $c$ is a $n$-dimensional column vector with its components either 0 or 1. For each $i$ with $1 \leq i \leq n$, $c_i = 1$ if $x_i$ is on the boundary of the region. Otherwise, $c_i = 0$. $\lambda c c^T$ is a penalty term that corresponds to the potential reference $(1)_4$, $f$ only depends on the current $j$ through the boundary of the region. In this study, the direct method from [13] is employed for the solution of this system.

In MCMC for EIT, forward map $u_E(x)$ has to be evaluated for a series of $\{x^{(i)}\}$. Normally, the stiffness matrix is formed for each $x^{(i)}$, then the resulting sparse system is solved by the Cholesky factorization. In this study, we obtain the Cholesky factor of the stiffness matrix $K^{i+1}$ for $x^{(i+1)}$ by modifying on that of $x^i$ as in [5]. Suppose that $x^{(i+1)}$ differs from $x^{(i)}$ at only its $k$-th component, and $x_k^{(i+1)} - x_k^{(i)} = \delta x$. Then, according to (5),

$$K^{(i+1)} = K^{(i)} + \delta x K_k. \tag{6}$$

Recall that $K_k$ is sparse, symmetric and of low rank. So Eq. (6) can be written as

$$K^{(i+1)} = K^{(i)} \pm C C^T \tag{7}$$

for some suitable $C$. Equation (7) indicates that the method in [5] can be employed to find the Cholesky factor of $K^{(i+1)}$ based on that of $K^{(i)}$. When $x^{(i+1)}$ differs from $x^{(i)}$ at more than one component, $x^{(i+1)}$ can be obtained from a series of successive one-component updates on $x^{(i)}$. Hence, the Cholesky factor of $K^{(i+1)}$ can be obtained from a series of updates on that of $K^{(i)}$.

Efficiency of our method of updating Cholesky factor was measured on the toy EIT model (Fig. 1). In FEM, the region was discretized into $24 \times 24$ grids of equal size. Bilinear rectangular element was employed, and the resulting local stiffness matrix was

$$K_4 = \frac{1}{3} \begin{bmatrix} 2 & -0.5 & -1 & -0.5 \\ -0.5 & 2 & -0.5 & -1 \\ -1 & -0.5 & 2 & -0.5 \\ -0.5 & -1 & -0.5 & 2 \end{bmatrix}.$$

Given that $u_E(x)$ had been computed, $u_E(x + \Delta x)$ was calculated by updating of Cholesky factor method as well as in the standard way. All the computation was carried out in MATLAB, and the CPU time was recorded with the 'profile' command. According to the record, updating Cholesky factor method was found to be about *five* times more efficient than the standard method, when $\Delta x$ had only one non-zero component. The method for updating Cholesky factor could still be four and three times more efficient when the non-zero component of $\Delta x$ was two and four. As $\Delta x$ had more non-zero components, updating Cholesky factor method became less efficient.

### 3.2. Approximate computation

In MCMC for EIT, the forward map of EIT is sometimes computed approximately with local linearisation. Approximate computation of forward map can be used to make efficient MCMC algorithms, because it is much cheaper than the exact computation. The approximation of forward map has to be as accurate as possible, in order for MCMC algorithms involving it to achieve

maximum efficiency. Because of this, we modify the ordinary approximation of forward map with local linearisation, and make it more accurate.

Normally, the approximation for $u_E(x + \Delta x)$ with local linearisation at $x$ is

$$u_E^*(x + \Delta x) \equiv u_E(x) + \left.\frac{\partial u_E}{\partial x}\right|_x \Delta x. \tag{8}$$

Here $\partial u_E/\partial x$ is the Jacobian of forward map. Computing $u_E^*(x+\Delta x)$ is well known to be much faster than $u_E(x + \Delta x)$. In this study, a new approximation is made by introducing a log transformation into (8). Define $\rho(x) = \log x$ and $\widehat{u}_E(\rho(x)) = u_E(x)$. Let $\Delta\rho = \log(x + \Delta x) - \log(x)$ so that $u_E(x + \Delta x) = \widehat{u}_E(\rho + \Delta\rho)$. Then,

$$\widehat{u}_E^*(\rho + \Delta\rho) \equiv \widehat{u}_E(\rho) + \left.\frac{\partial \widehat{u}_E}{\partial \rho}\right|_\rho \Delta\rho \tag{9}$$

is our new approximation to $u_E(x + \Delta x)$.

The accuracy of the two approximations (8) and (9) was compared on our toy model (1), in terms of the log likelihood term: $L(x + \Delta x) = -\frac{1}{2\varsigma^2}\|u_E^{\text{noise}} - u_E(x + \Delta x)\|^2$. The approximated log likelihood term $L^*(x + \Delta x) = -\frac{1}{2\varsigma^2}\|u_E^{\text{noise}} - \dot{u}_E^*(x + \Delta x)\|^2$ was computed with $\dot{u}_E^*(x + \Delta x)$ being $\widehat{u}_E^*(x + \Delta x)$ and $u_E^*(x + \Delta x)$, respectively. The error $\Delta L = \|L^*(x + \Delta x) - L(x + \Delta x)\|$ of approximated likelihood term, associated with approximation (8) and (9) was then compared.

In practice, $\Delta L$ was computed for each $\Delta x = (\xi, \xi, \cdots, \xi)$ with $\xi$ taking various values around zero. $\Delta L$ was plotted against $\xi$ in Fig. 2. In this figure, the dashed and continuous curves correspond respectively to the ordinary local linearisation (8) and the local linearisation with log transformation (9). As shown, the continuous curve is always well below the dashed one. This indicates that our approximation (9) is more accurate.

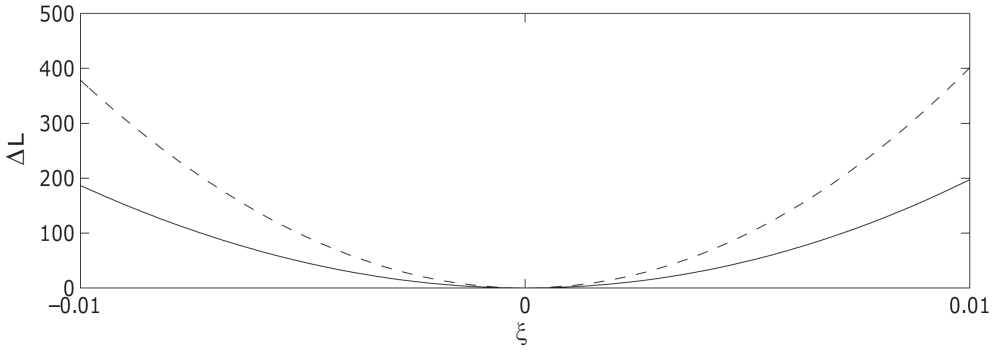


**Fig. 2.** Error $\Delta L$ of approximated log likelihood based on two approximations of forward map $u_E(x + \Delta x)$ when $\Delta x = (\xi, \xi, \cdots, \xi)$. The continuous and dashed curves correspond to the local linearisation with and without log transformation, respectively.

## 4. Novel mcmc algorithms

### 4.1. Multiple prior block update

Our first novel MCMC algorithm is written in the 'multiple prior updates, likelihood acceptance ratio' scheme [6]. This scheme is a special case of the surrogate transition method [2]. Assume $\pi(x) = \pi_p(x)\pi_l(x)$ is the target distribution to be sampled. The multiple prior update scheme is the surrogate transition method with $\pi_p(x)$ as its global approximation $\pi^*(x)$ of $\pi(x)$. Assume that the surrogate Markov chain begins with state $x$ and ends with state $y$. $S(x, y)$ is the transition

function from $x$ to $y$. Then the probability for the state $y$ to be accepted in the main Markov chain converging to $\pi(x)$ is

$$\min\left\{1, \frac{\pi(y)S(y,x)}{\pi(x)S(x,y)}\right\} = \min\left\{1, \frac{\pi_p(y)\pi_l(y)S(y,x)}{\pi_p(x)\pi_l(x)S(x,y)}\right\} = \min\left\{1, \frac{\pi_l(y)}{\pi_l(x)}\right\}. \tag{10}$$

The second equation in (10) is because

$$\pi_p(y)S(y,x) = \pi_p(x)S(x,y). \tag{11}$$

i.e., the surrogate chain is a reversible Markov chain converging to $\pi_p$. Equation (10) indicates that the proposal $y$ almost always gets accepted when $\pi_l(y) \approx \pi_l(x)$.

In this study, the update in the surrogate Markov chain is made on a block of neighboring components altogether, as suggested by [14]. However, the block update here leaves the mean value of state variable invariant. Assume that $x$ and $x'$ are the current and proposal state, respectively. The indices for all the components in the $i$-th block is $b^i = \left(b_1^i, b_2^i, \cdots, b_m^i\right)$. The update at this $i$-th block is

$$x'_{b^i} = x_{b^i} + (I_m - a \cdot a^t)\varepsilon. \tag{12}$$

Here $x_{b^i}$ and $x'_{b^i}$ are the components of current and proposal state that are in the $i$-th block, respectively, $\varepsilon \sim N(0, \sigma^2 I_m)$ and $a = (1/\sqrt{m}, 1/\sqrt{m}, \cdots, 1/\sqrt{m})$. As shown $(I_m - a \cdot a^t)\varepsilon$ has zero mean, so $x'$ and $x$ have the same mean value. Since $x$ and $x'$ also differ from each other at only few neighbouring components, $u_E(x) \approx u_E(x')$. Hence $\pi_l(x) \approx \pi_l(x')$. So proposal $x'$ will very likely get accepted according to (10).

In practice, the multiple prior block update is mixed with single-site Metropolis ( [8]) when sampling the posterior distribution (2). The block update (12) maintains the mean value of the state variable, and the Markov chain updated by this proposal alone is reducible. After Inclusion with single-site update, the Markov chain will be desirably irreducible [2]. In practice, 90% of the moves in this mixture are the multiple prior update, so that this mixture update inherit most of its characteristic. The block was chosen at random each time, and the blocks can overlap each other. This algorithm based on the $2 \times 2$ block was found to perform the best. The optimal update in the surrogate chain was found to be about 10.

The performance of our multiple prior block update is compared with that of the single-site Metropolis [8]. As in [3], Fig. 3 shows the traces of three representative pixels in the conductivity image from these two algorithms. As seen, the three pixels mixed much better in our multiple prior block update. This indicates that our multiple prior block update performs much better. The two algorithms are also compared in terms of the mixing property of the middle curve in the three representative pixels. As stated in [3], the middle pixel is on the boundary of conductivity image, and transverse from low to high value during the MCMC run. Its mixing property is an extremely important indicator for the performance of MCMC algorithm.

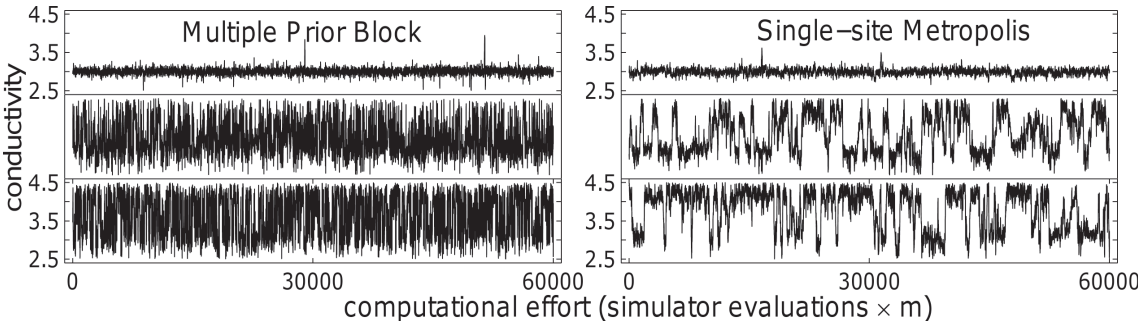


**Fig. 3.** Traces of three representative pixels from Markov chains updated by multiple prior block update and single-site Metropolis.

In case of multiple prior block update, it took $10 \times 10 \times \mathrm{m}$ updates for the autocorrelation of the blue pixel to go down to 10%, while it took 10 times more updates in case of the single-site Metropolis. On the other hand, the CPU time for one multiple prior block update is about 1.7 times more than that of a single-site update. Overall, the multiple prior block update is about $10/1.7 \approx 6$ times more efficient than the single-site Metropolis.

### 4.2. Delayed-acceptance Metropolis-Hastings

Our second efficient MCMC algorithm is constructed in the delayed-acceptance Metropolis-Hastings (DAMH) scheme [7]. As the surrogate transition method, the DAMH also uses an auxiliary probability distribution to make the proposal move. The auxiliary distribution in DAMH can however be a state-dependent approximation of the target distribution. This makes DAMH distinctive.

Assume that $\pi(.)$ is the target distribution to be sampled. The Markov chain to be designed to converge to $\pi$ is currently at the state $x$. Depending on $x$, $\pi_x^*(.)$ is an approximation of $\pi(.)$. With the terminology from [2], assume that $T_1(x, y)$ is an initial transition function for the Markov chain.

In Metropolis-Hastings (M-H) scheme [11], the proposal $y \neq x$ from $T_1(x, y)$ has to be tested in order to be accepted as the state for the next step. The test involves an acceptance probability $\alpha$ and is: draw a sample from $r \sim U(0, 1)$; if $r < \alpha$, accept the proposal. Otherwise, reject the proposal and the current state remains to be the state of the next step.

In DAMH, the proposal $y$ has to be subject to *two* steps of the above tests, in order to become the state for the next step. The two acceptance probabilities in the two tests are

$$\alpha_1 = \min\left\{1, \frac{T_1(y,x)\pi_x^*(y)}{T_1(x,y)\pi_x^*(x)}\right\}, \qquad \text{and} \qquad \alpha_2 = \min\left\{1, \frac{T_2(y,x)\pi(y)}{T_2(x,y)\pi(x)}\right\}, \tag{13}$$

respectively. Here

$$T_2(x,y) = T_1(x,y)\min\left\{1, \frac{T_1(y,x)\pi_x^*(y)}{T_1(x,y)\pi_x^*(x)}\right\}, \qquad y \neq x. \tag{14}$$

The actual transition function based on the initial transition function $T_1(x, y)$ is:

$$A(x,y) = T_1(x,y)\underbrace{\min\left\{1, \frac{T_1(y,x)\pi_x^*(y)}{T_1(x,y)\pi_x^*(x)}\right\}}_{\text{step}I}\underbrace{\min\left\{1, \frac{T_2(y,x)\pi(y)}{T_2(x,y)\pi(x)}\right\}}_{\text{step}II}, \qquad y \neq x. \tag{15}$$

Assume that

$$\frac{\pi_x^*(y)}{\pi_x^*(x)} \approx \frac{\pi(y)}{\pi(x)} \tag{16}$$

for all $x$ and $y$ that are close to each other. Then,

$$\alpha_1 \approx \min\left\{1, \frac{T_1(y,x)\pi(y)}{T_1(x,y)\pi(x)}\right\}, \qquad \alpha_2 \approx 1. \tag{17}$$

Equation (17) indicates that the acceptance probability in the first step of DAMH is approximately the same as the that of M-H given the same initial transition function. The actual transition function in DAMH is effectively almost the same as that of M-H given the same initial transition function. In other words, DAMH produces almost the same amount of statistically independent samples as M-H, given the same initial transition function.

When the target distribution $\pi(.) = \pi(.|u_E^{\text{noise}})$ in (2), a candidate of $\pi_x^*(.)$ is

$$\pi_x^*(z|u_E) \propto \exp\left\{\beta \sum_{i\sim j} v(z_i - z_j)\right\} I_{[2.5,4.5]^n}(z) \cdot \exp\left\{-\frac{1}{2\varsigma^2}\|u_E^{\text{noise}} - u_E^*(z)\|^2\right\}. \tag{18}$$

Here $u_E^*(z)$ is the local linearisation of the forward map at state $x$, as discussed in Subsec. 3.2. Apparently, (18) satisfies the assumptions (16). Furthermore, evaluating $\pi_x^*(y)/\pi_x^*(x)$ has negligible cost compared to that of $\pi(y)/\pi(x)$. When computing $\pi(y)/\pi(x)$ takes up almost all the CPU time for each update in MH and DAMH, the average CPU time for each DAMH update is $\alpha_1$ times that of M-H, given the same initial transition function. This is because $\pi(.)$ is only required to be evaluated $\alpha_1$ times on average in each DAMH update; but every time in each M-H update. In DAMH, evaluating $\pi(.)$ only happens in computing $\alpha_2$, which is only required when the proposal gets accepted in the first step. Recall that $\alpha_1$ is the probability for the proposal to get accepted in the first step. Therefore, $\pi(.)$ is computed $\alpha_1$ times on average in each DAMH update. Overall, the same initial transition function in DAMH is expected to be $1/\alpha_1$ times more efficient than in M-H.

In this study, DAMH with single-site update was discussed for sampling the posterior distribution (2) of interest.

First, our DAMH algorithm used two different approximations of forward map, as discussed in Subsec. 3.2. The performance of the DAMH with these two approximations was then compared. Figure 4 shows the empirical acceptance rate at the second step of DAMH for the two approximations of forward map. As seen, the second step of DAMH is close to 1 as expected in case of approximation with log transformation; while it is well below 1 in case of approximation with ordinary local linearisation. Therefore, approximated forward map with log transformation is a better approximation and enables the DAMH algorithm to perform better.

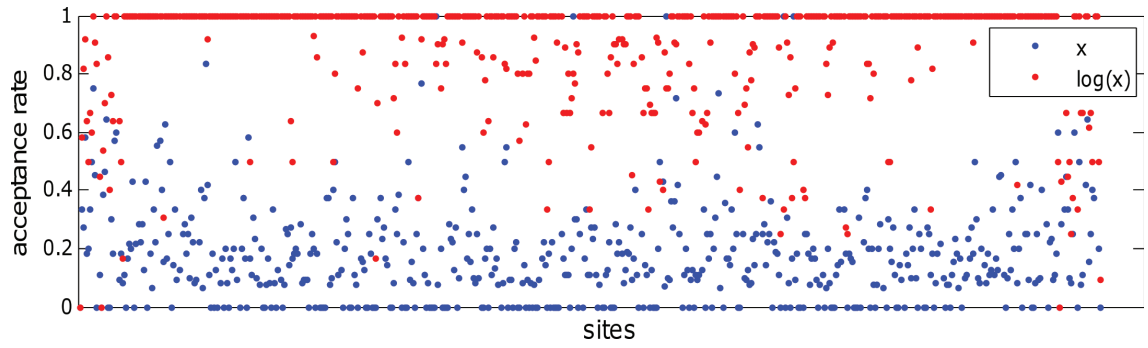


**Fig. 4.** Empirical acceptance rates in the second step of DAMH with two approximations of forward maps.

Second, DAMH with single-site update and single-site Metropolis (single-site update in M-H) are compared on sampling the posterior distribution (2) of interest. Figure 5 shows the empirical acceptance rates at every site for these two algorithms: As seen, the acceptance rates at the second step of DAMH are all close to 100%; while the ones in the first step of DAMH are all close to the acceptance rate of single-site Metropolis (single-site update in M-H). This finding indicates that the

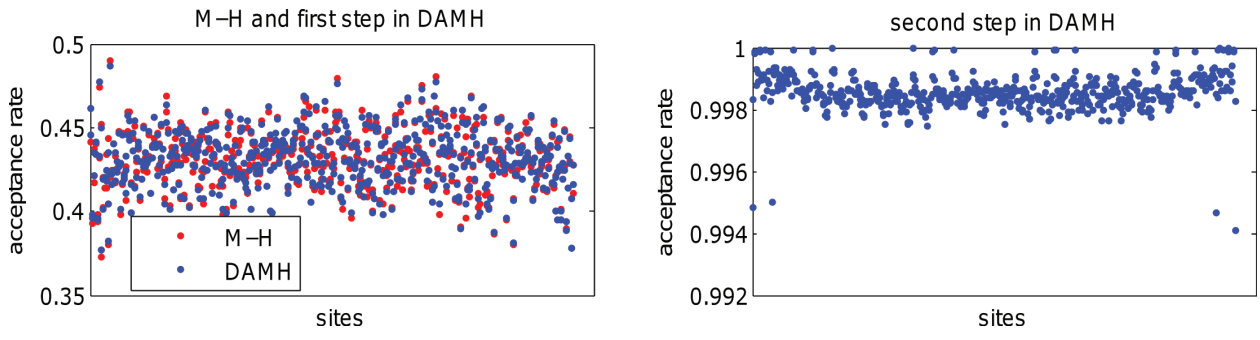


**Fig. 5.** Empirical acceptance rate at every site in DAMH with single-site update and single-site Metropolis.

DAMH with single-site update has achieved the expected performance. In other words, single-site update is about two times more efficient in DAMH than in M-H.

## 5. Conclusion

In Sec. 3, we speed up the exact computation of forward map of EIT, by updating the Cholesky factor of the stiffness matrix [5]. With regard to the toy EIT model, our method of updating Choleksy factor can be as much as *five* times more efficient than the standard computation. We also make the approximation of forward map with local linearisation more accurate, by introducing a log transformation into this process.

In Sec. 4, we present two novel and efficient MCMC algorithms that perform more efficient than the single-site Metropolis for sampling a posterior distribution of our toy EIT model. One algorithm is made in the multiple prior update scheme. The surrogate chain in this scheme is updated by the constrained block update that leaves the mean value of state variable invariant. This multiple block update was found to be *six* times more efficient than the standard single-site Metropolis. The other MCMC algorithm is DAMH with single-site update. This algorithm was found to be twice more efficient than the standard single-site Metropolis.

As in [3], this study discusses MCMC sampling for a posterior distribution in the Bayesian approach to EIT. This study equipped single-site Metropolis with efficient forward map solver, on condition that the sparse system involved is solved by the direct method. We also present two novel and efficient MCMC algorithms that are more efficient than single-site Metropolis, which is considered hard to surpass [3].

On the other hand, the sparse system in the forward map computation of EIT is solved by the direct method in this paper. As EIT model becomes larger and more practical, the resulting sparse system will be larger, and the iterative method will be more suitable. Therefore, it is important to study forward map computation of EIT with iterative method for the sparse system. Furthermore, the proposal moves in our efficient MCMC algorithms are still limited to a few components update. Future work should also focus on devising efficient MCMC algorithms with multi-variant update.

## Acknowledgements

The author would like to thank his former PhD surpervisor for his support on this study. This work is also funded by the Research Development Fund from Xi'an Jiaotong Liverpool University.

## References

[1] C. Fox, G. Nicholls. Sampling conductivity images via MCMC. In *Proc. of Leeds Annual Stat. Research Workshop*, pp. 91–100, 1997.

[2] J.S. Liu. *Monte Carlo strategies in scientific computing*. Springer, 2001.

[3] D. Higdon, C.S. Reese, J.D. Moulton, J.A. Vrugt, C. Fox. *Posterior exploration for computationally intensive forward models*. In S. Brooks, A. Gelman, G.L. Jones, X. Meng [Eds.], *Handbook of Markov chain Monte Carlo*, CHAPMAN & HALL/CRC, pp. 401–418, 2011.

[4] A. Ern, J. Guermond. *Theory and practice of finite elements*. Springer, 2004.

[5] T.A. Davis, W.W. Hager. Multiple-rank modifications of a sparse Cholesky factorization. *SIAM Journal on Matrix Analysis and Applications*, **22**(4): 997–1013, 2001.

[6] G.K. Nicholls, C. Fo. Prior modeling and posterior sampling in impedance imaging. *Proc. SPIE*, **3459**: 116–127, 1998.

[7] J.A. Christen, C. Fox. MCMC using an approximation. *Journal of Computational and Graphical Statistics*, **14**(4): 795–810, 2005.

[8] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, **21**: 1087–1091, 1953.

[9] M. Cheney, D. Isaacson, J.C. Newell. Electrical impedance tomography. *SIAM Review*, **41**(1): 85–101, 1999.

[10] D.S. Holder. *Electrical impedance tomography: methods, history and applications*, Institute of Physics Publishing, Bristol, UK, 2005.
[11] W.K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, **57**(1): 97–109, 1970.
[12] W.S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of Computational Statistical Association*, **74**: 829–836, 1979.
[13] T.A. Davis. *CHOLMOD Version 1.0 User Guide*. Dept. of Computer and Information Science and Engineering, University of Florida, 2005.
[14] W.R. Gilks, S. Richardson, D.J. Spiegelhalter. *Markov chain Monte Carlo in practice*. CHAPMAN & HALL, 1996.